Multisample-Based Contrastive Loss for Top-K Recommendation

Hao Tang, Guoshuai Zhao, *Member, IEEE*, Yuxia Wu, and Xueming Qian, *Member, IEEE*

***Abstract*—Top-k recommendation is a fundamental task in recommendation systems that is generally learned by comparing positive and negative pairs. The contrastive loss (CL) is the key in contrastive learning that has recently received more attention, and we find that it is well suited for top-k recommendations. However, CL is problematic because it treats the importance of the positive and negative samples the same. On the one hand, CL faces the imbalance problem of one positive sample and many negative samples. On the other hand, there are so few positive items in sparser datasets that their importance should be emphasized. Moreover, the other important issue is that the sparse positive items are still not sufficiently utilized in recommendations. Consequently, we propose a new data augmentation method by using multiple positive items (or samples) simultaneously with the CL loss function. Therefore, we propose a multisample-based contrastive loss (MSCL) function that solves the two problems by balancing the importance of positive and negative samples and data augmentation. Based on the graph convolution network (GCN) method, experimental results demonstrate the state-of-the-art performance of MSCL. The proposed MSCL is simple and can be applied in many methods. Our code is available at https://github.com/haotangxjtu/MSCL.**

***Index Terms*—Contrastive loss, recommendation system, data augmentation, graph convolution network.**

I. Introduction

Recommendation systems have become an important research field that aims to solve the information overload problem in the information explosion era. Recommendation systems are widely used in many fields, such as e-commerce [1], [2], life services [3]–[5], social networks [6], [7], and entertainment [8], [9], and they have become one of the important technologies in the information age. Top-k recommendation is the basic problem of recommendation systems that learns the users' preferences through their historical interaction records. Then, it recommends the top-k items to the users that they may like.

Deep learning-based top-k recommendation algorithms significantly improve recommendation performance and have become the mainstream research direction in recent years, especially collaborative filtering-based methods. These existing algorithms extract advanced semantic features and perform complex feature interactions by employing MLP [10], CNN [11], [12], RNN [13], attention mechanisms [14], [15], etc. The user-item interaction is naturally viewed as a bipartite graph. Graph convolutional network (GCN)-based methods are increasingly integrated with recommendation systems [16]–[18]. For example, a hierarchical user intent graph network (HUIGN) [17] exhibits user intents in a hierarchical graph structure from fine-grained to coarse-grained intents. The multimodal graph convolution network (MMGCN) [18] leverages information interchange between users and items to enhance user representations and further capture users' fine-grained preferences on different modalities. GCN-based methods aggregate features of neighbors as well as higher-order neighbors to obtain better feature representations of users and items, and the performance is further improved.

In contrast to the rapid development of recommendation methods, the loss function has rarely been improved. There are many loss functions for the recommendation, such as mean square loss (MSE), cross entropy loss, Bayesian Personalized Ranking (BPR) [19], and so on. MSE is always used for rating prediction [20]–[22], and when it comes to the top-k recommendation, the last two loss functions are usually used [10], [23], [24]. Cross entropy loss treats the top-k task as a classification problem while BPR treats it as a ranking problem which encourages the ranking of positive items above negative items for the given user. BPR is more suitable for the top-k recommendation, so it becomes the most popular and widely used loss function. Recently, the contrastive loss (CL) function has yielded excellent results in several fields under the contrastive learning framework [25]–[30]. CL directly treats nonpositive items within the same training batch as negative samples, while BPR uses one or several negative samples with additional sampling time. Thus, CL can obtain a large number of negative samples simply and quickly. They all

Manuscript received 22 August 2021; revised 29 October 2021; accepted 1 November 2021. Date of publication 9 November 2021; date of current version 7 February 2023. This work was supported in part by NSFC, China, under Grants 61902309, 61701391, and 61772407, in part by ShaanXi Province under Grant 2018JM6092, in part by the Fundamental Research Funds for the Central Universities, China under Grant xxj022019003, in part by China Postdoctoral Science Foundation under Grant 2020M683496, and in part by National Postdoctoral Innovative Talents Support Program, China, under Grant BX20190273. The Associate Editor coordinating the review of this manuscript and approving it for publication was Prof. Liqiang Nie. *(Corresponding author: Guoshuai Zhao.)*

Hao Tang and Yuxia Wu are with the School of Information and Communication Engineering, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: th1002@stu.xjtu.edu.cn; wuyuxia@stu.xjtu.edu.cn).

Guoshuai Zhao is with the School of Software Engineering, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: guoshuai.zhao@xjtu.edu.cn).

Xueming Qian is with the Ministry of Education Key Laboratory for Intelligent Networks and Network Security, School of Information and Communication Engineering, and SMILES LAB, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: qianxm@mail.xjtu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TMM.2021.3126146.

Digital Object Identifier 10.1109/TMM.2021.3126146

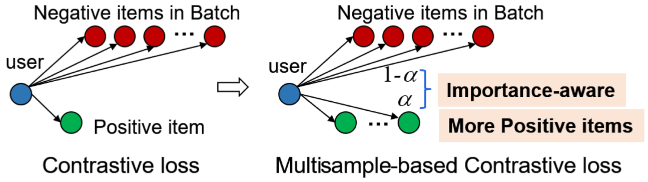


Fig. 1. We propose a multisample-based contrastive loss (MSCL) that distinguishes the importance of positive and negative samples and makes better use of sparse positive samples by a new data augmentation method.

learn through a contrastive process, so BPR loss can also be seen as a kind of contrastive loss.

However, the importance of positive and negative samples should be treated differently by CL. (1) CL uses one positive sample and N-1 negative samples, where N is the batch size, typically 1024, 2048, etc. Thus, the imbalance problem or different importance values of one positive sample and negative samples should be addressed. Contrastive learning often adopts the contrastive loss function for optimization which also suffers from this problem. (2) The number of positive samples is very small; thus, recommendation systems face the sparsity problem. Intuitively, the sparser the dataset, the fewer positive samples, and the more important the positive samples should be relative to many negative samples. Therefore, positive and negative samples should be treated differently for the above two reasons. As large negative samples can help the model to learn discriminative features, we solve this problem based on this. It is this distinction that allows the roles of positive and negative samples to be appropriately adjusted and makes them work better collaboratively.

Another issue we are concerned about is the insufficient use of positive samples in top-k recommendations. As mentioned before, there are very limited positive items of each user in the recommendation system. How to make full use of the existing positive samples is a key problem. Data augmentation methods help to solve this problem. Data augmentation methods in recommendation systems are generally based on graph structures, such as edge or node dropout, masking features, and random walks. The potential of the combined use of positive samples is not exploited.

To solve the above problems, we propose a new CL-based loss function, and the basic idea is shown in Fig. 1. For the first problem, we distinguished their different importance values by adjusting the weights of positive and negative samples. The hyperparameter α is the weight of the positive samples, which represents their importance. To make better use of positive items, we propose a new data augmentation method by using multiple positive samples simultaneously. This data augmentation makes better use of positive samples and the training space can be expanded because of different combinations of multiple positive samples. In the original situations, the number of items the user has interacted with can be interpreted as the number of cases the user can encounter. By a random combination of multiple items, the user can encounter more cases, thus expanding the training space. Moreover, this data augmentation can be used for many other types of data, not just graph data.

In summary, we propose a contrastive loss function based on multiple (positive and negative) samples, which is named multisample-based contrastive loss (MSCL). The proposed loss function can significantly improve the performance and the training efficiency of the top-k recommendation with almost no increase in complexity. As shown in this paper, MSCL can be widely used for recommendations in various fields, such as Yelp's restaurant recommendations, Amazon's book recommendations, Alibaba's fashion recommendations. And MSCL makes basic method MF more competitive and is suitable for industrial applications at a large scale. The main contributions of this paper are summarized as follows:

- We propose a simple but effective loss function, MSCL, which improves the contrastive loss to make it suitable for the recommendation system. MSCL can be applied to many recommendation methods and is much better than the traditional BPR loss.
- The MSCL function distinguishes the importance of positive and negative samples by weighting. It helps to address the imbalance problem of positive and negative samples and to enhance the importance of positive samples in sparser datasets.
- We propose a new data augmentation method by using multiple positive samples simultaneously, which makes better use of the positive samples.
- Experimental results demonstrate state-of-the-art performance and many other advantages, such as broad applicability and high training efficiency. MSCL is suitable for top-k recommendations, and it makes the simple and basic MF more competitive.

The rest of this paper is organized as follows. In Section II, related works are briefly reviewed. To verify the effectiveness of MSCL, we design sLightGCN_MSCL, which combines MSCL with the best baseline as our method in Section III. Experiments and discussions are described in Section IV. Section V discusses the advantages of MSCL with more experiments. Conclusions are drawn in Section VI.

II. Related Work

In this section, we briefly review related works: contrastive loss and graph data augmentation methods. Differences between our approach and existing works are also presented.

A. Contrastive Loss (CL)

Contrastive loss has become an excellent tool in unsupervised representation learning with the development of unsupervised learning [31], [32]. It aims to maximize the similarities of positive pairs and minimize that of negative pairs [25], [26], [33]–[35]. CL is widely used for many kinds of data, such as images, text, audio, graphs, etc. It has been applied in the field of recommendation [36], [37].

Broadly, functions that use pairwise contrastive learning processes are contrastive loss functions that have many forms. BPR and triplet loss are the basic contrast-based and widely used loss functions. BPR [19] loss aims to maximize the distance between the positive pair and negative pair, which is proposed for the ranking task and widely used in top-k recommendations. Triplet loss [38]–[40] can be used to train samples with small differences, especially for human faces. The samples are triplets (anchor, positive, negative). Triplet loss is calculated by optimizing the distance between the anchor and positive samples so that it is smaller than the distance between the anchor and negative samples.

However, they employ limited pairs of samples. Contrastive loss functions based on multiple pairs of samples are more efficient and contain multiclass N-pair loss, InfoNCE loss, nonparametric softmax classifiers, and normalized temperature-scaled cross entropy loss (NT-Xent loss). Multiclass N-pair loss [41] is proposed from a deep metric learning perspective, which greatly improves the triplet loss by jointly pushing out multiple negative samples at each update. InfoNCE loss [42]–[44] is proposed by maximizing a lower bound on mutual information based on noise-contrasting estimation. The nonparametric softmax classifier [45] is presented by maximizing the distinction between instances via a novel nonparametric softmax formulation in an unsupervised feature learning approach. They come from different fields and formula derivations but share a similar form. NT-Xent loss [25], [30] is proposed on these bases, but with the minor difference that the denominator does not contain positive samples. All of them are widely used in contrastive learning frameworks and always obtain state-of-the-art results.

Contrastive loss has recently been used in recommendation systems in the contrastive learning framework for recommendations. For example, many works [37], [46], [47] have been proposed for sequential recommendation, and CLCRec [48] has been proposed for cold-start recommendation. SGL [36] was proposed for top-k recommendation, but it did not improve CL to fit the recommendation field.

B. Graph Data Augmentation

The user-item interaction records in recommendation systems are naturally viewed as bipartite graphs. Graph-based data augmentation is widely used and studied in graph contrastive learning, which contains both traditional subgraph sampling methods and recently proposed methods [30], [36], [49]–[52]. Users and items are inherently linked and dependent on each other in the user-item bipartite graph. Data augmentation for GCNs is also challenging due to the complex, non-Euclidean structure of the graph, and few works study the data augmentation of graphs. Therefore, graph data augmentation must be tightly integrated with the graph rather than replicating the methods used in the computer vision and natural language processing domains.

Graph data augmentation conforms to the basic assumptions of graph data processing. Node dropping assumes that a missing edge vertex does not alter semantics. Edge perturbation is considered to improve the robustness of the semantics against connectivity changes. Masking node features enhances semantic robustness by losing some attributes for each node. Subgraphs assume that local structure can hint at the complete semantics [30].

Graph augmentation can be divided into two types, feature-space augmentations and structure-space augmentations: (1) feature-space augmentations are realized by modifying initial node features, such as masking features or adding Gaussian noise, and (2) structure-space augmentations operate on the graph structure by adding or removing nodes or edges (edge perturbation), subsampling or subgraphs by random walk, or generating different views using shortest distances or diffusion matrices [51].

Recently, Zhu *et al.* [52] proposed adaptive graph augmentation to design augmentation schemes that tend to keep important structures and attributes unchanged while perturbing the unimportant links and features. Zhao *et al.* [53] utilized a neural edge predictor to predict likely edges for graph augmentation to improve node classification performance. For top-k recommendations, the latest work, SGL [36], uses three operators on the graph structure, namely, node dropout, edge dropout and random walk. The experimental results show that edge dropout performs the best.

C. Differences With Existing Works

Differences with existing CL functions; many works are done using only the CL function in the contrastive learning framework. SGL in the recommendation system employs a multitasking mechanism with joint use of CL and BPR. Despite some improvements proposed on CL, such as soft contrastive loss [54] and debiased contrastive loss [55], they all treat the weights of the positive samples and negative samples as the same. The problem of imbalanced positive and negative samples is still not a concern. More importantly, how to adapt CL to recommendation systems is a new topic worth investigating, especially to emphasize the importance of positive samples in sparser datasets. Thus, the proposed importance-aware CL is different from previous works.

Differences with existing graph data augmentation: Existing graph data augmentations in recommendation systems are common methods in the graph field. How to make full use of the limited positive samples to obtain better results, especially for the recommendation system, is an important task and challenge for data augmentation. We randomly sampled a fixed number within one-hop neighbors that was not the same as random dropout or a subgraph by random walk on multiple hops. More importantly, our method is a structure-space-based augmentation, and traditional structure-space-based methods generally work in the aggregation process of GCNs. Traditional augmented data are used one by one under the same loss, which is a serial approach. We use multiple positive samples at the same time and combine them explicitly with the loss function, which is a parallel method for better constraints.

III. Methodology

We designed a method named sLightGCN_MSCL that combines MSCL with a strong baseline, as shown in Fig. 2, to verify the effectiveness of the proposed loss function. In this section, we first briefly describe the basic methods of LightGCN, then focus on the MSCL function, and finally is the model analyses.

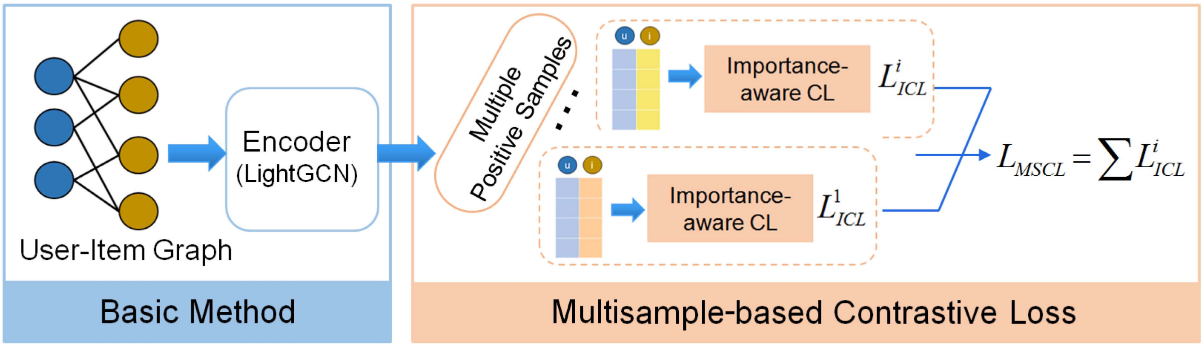


Fig. 2. An illustration of our method. Many models can be used as the encoder, and LightGCN is used here as an example.

A. Basic Method

Recently, graph-related methods have shown excellent performances, which treat user-item interactions as graph structures and adopt graph convolution networks. Combined with collaborative filtering, NGCF [23], LR-GCCF [24], LightGCN [56], etc., are excellent models for top-k recommendation.

LightGCN is a state-of-the-art method and is introduced here as our main baseline. This model includes only the most essential component in GCN - neighborhood aggregation - for collaborative filtering, which is much easier to implement and train and gains substantial improvements. Then, neighborhood aggregation is defined as follows:

$$\boldsymbol{e}_u^{(k+1)} = \sum_{i\in\mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_i|}}\boldsymbol{e}_i^{(k)} \quad (1)$$

$$\boldsymbol{e}_i^{(k+1)} = \sum_{u\in\mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_i|}\sqrt{|\mathcal{N}_u|}}\boldsymbol{e}_u^{(k)}. \quad (2)$$

where u, i denote the user and the item in the user-item graph, and $e_u^{(k)}, e_i^{(k)}$ denote embeddings of u, i of the k-th layer. Specifically, $k = 0$ represents the initialized latent vector; $\mathcal{N}_u$ and $\mathcal{N}_i$ represent the set of neighbors of targets u and i, respectively. The final embeddings of users and items are:

$$\boldsymbol{e}_u = \sum_{k=0}^{K} \alpha_k \boldsymbol{e}_u^{(k)} \quad (3)$$

$$\boldsymbol{e}_i = \sum_{k=0}^{K} \alpha_k \boldsymbol{e}_i^{(k)} \quad (4)$$

where K is the number of layers; α_k denotes the importance of the k-th layer embedding, and they can be treated as a hyperparameter to be tuned manually or as a model parameter to be optimized automatically. Following the original paper of LightGCN, the mean of embeddings from all layers are adopted as the final embeddings, that is, α_k=1/(K+1).

LightGCN-single, a variant of LightGCN, is also proposed in the original paper [56], where only the k-th embeddings, $e_u^{(k)}, e_i^{(k)}$, are used as final embeddings. This equals to tune α_k rather than simply set it as 1/(K+1) uniformly. It has shown better performance than LightGCN on many datasets. Therefore, it is selected as a strong baseline in order to get the best results here and is named **sLightGCN** for short.

The BPR loss is used for training in LightGCN. We present it here for comparison with MSCL:

$$L_{BPR} = \sum_{(u,i,j)\in D} -\log\sigma\left(\hat{y}_{ui} - \hat{y}_{uj}\right) \quad (5)$$

where $D = \{(u, i, j), u \in U, i, j \in I\}$, U, and I are the set of users and items, i, j denote positive and negative items, respectively, $\sigma(\cdot)$ is the logistic sigmoid function, and $\hat{y}_{ui}$ is the inner product of the user and item, which is the same as Eq. (12).

B. Multisample-Based Contrastive Loss

1) The Basic Contrastive Loss: Referring to some recent works [25], [30], we use NT-Xent as the original contrastive loss function and then adapt it to the recommendation field.

The NT-Xent is:

$$L = -\log\frac{\exp\left(\operatorname{sim}\left(\boldsymbol{z}_i, \boldsymbol{z}_j\right)/\tau\right)}{\sum_{k=1,k\neq i}^{N}\exp\left(\operatorname{sim}\left(\boldsymbol{z}_i, \boldsymbol{z}_k\right)/\tau\right)} \quad (6)$$

where $sim(\boldsymbol{z}_i, \boldsymbol{z}_j) = \boldsymbol{z}_i^\top \boldsymbol{z}_j/\|\boldsymbol{z}_i\|\|\boldsymbol{z}_j\|$, and z_i and z_j indicate the embeddings of sample i, j in a minibatch, N is the batch size, and τ denotes the temperature parameter. To fit the recommendation domain, we rewrite it as L_{CL}:

$$f(u, i) = \boldsymbol{e}_u^\top \boldsymbol{e}_i / \|\boldsymbol{e}_u\|\,\|\boldsymbol{e}_i\| \quad (7)$$

$$\begin{aligned} L_{CL} &= -\frac{1}{N}\sum_{(u,i)\in D}\log\frac{\exp\left(f(u,i^+)/\tau\right)}{\sum_{i\in I^-}\exp\left(f(u,i)/\tau\right)} \\ &= -\frac{1}{N}\sum_{(u,i)\in D}\left(f(u,i^+)/\tau - \log\sum_{i\in I^-}\exp\left(f(u,i)/\tau\right)\right) \end{aligned} \quad (8)$$

where $D = \{(u, i), u \in U, i \in I\}$ is a training batch; U and I are the sets of users and items, respectively; i^+ is the positive sample of target user u; and I^- is the set of negative samples. $f(u, i)$ is the cosine similarity of the (u, i) pair based on their embeddings. We follow the sampling strategy used in [25], [30] in which the other nonpositive samples in the same batch are seen as negative samples.

2) Importance-Aware CL (ICL): In the basic contrastive loss, the minus sign is preceded by one positive sample, followed by the sum of N-1 negative samples, which results in imbalance problems. In addition, emphasizing the importance of positive samples on sparser datasets is also a problem we want to address. These two problems can be solved together by weighting, an effective and common practice. The importance of positive and negative samples can be adjusted, which helps to better backpropagate and make the training more effective. We name the modified CL importance-aware CL:

$$L_{ICL} = -\frac{1}{N}\sum_{(u,i)\in D}\Big(\alpha f(u,i^{+})/\tau \\ -(1-\alpha)\log\sum_{i\in I^{-}}\exp\left(f(u,i)/\tau\right)\Big) \tag{9}$$

where α is the weight of the positive samples, and $\alpha \in [0,1]$.

When $\alpha = 0.5$, the L_{ICL} is the same as L_{CL}. Because problems in the analysis are inevitable, $\alpha = 0.5$ is not optimal in general. When $\alpha < 0.5$, it means that N-1 negative samples need more weights and relatively more losses to be optimized. When $\alpha > 0.5$, the positive samples need to be given more attention, which may be because there are too few positive items for users in the recommendation system. The weighting method is simple yet effective in adapting to a variety of datasets.

3) Multiple Positive Sample-Based CL (MCL): To address the problem that positive samples are insufficiently used, we propose a new data augmentation method that uses multiple positive samples simultaneously. We propose a multipath-based method to use multiple positive samples under the supervision of the CL function. The conventional approach is to use a random batch of data and a loss function to form a learning path after the model is computed. We extend this idea by randomly sampling M positive samples to form M paths. The target user is optimized by M positive samples. The final loss is the sum of M loss functions, which is calculated simply and effectively. Thus, the multiple positive sample-based CL is:

$$L_{MCL} = \sum_{m=1}^{M} L_{CL}^{m} \tag{10}$$

where M is a hyperparameter, which is the number of used positive samples.

This data augmentation comes with many benefits. (1) We keep the same training process of MSCL as the original one, but the difference is the number of samples used for each training time. Suppose the user has L positive samples, there are C_L^1 (combination formula) possible cases by random sampling for the user at each training time in the original way. MSCL uses M positive samples simultaneously for the user, so there are C_L^M possible cases for each training time. Therefore, the M positive samples greatly increase the cases that users can encounter. This provides augmentation and better usage of the existing positive items. (2) Positive and negative samples form a comparison; thus, the expanded positive samples also enlarge the comparable cases. (3) Furthermore, we integrate this augmentation with the loss function explicitly in parallel to facilitate increasingly better constraints and backpropagation for the user. (4) This data augmentation method can be widely applied to graph data as well as various other types of data.

4) Combining ICL and MCL as MSCL: We have elaborated on our two improvements, ICL and MCL. These two improvements are proposed from different perspectives. Combining them together can solve the two problems for the top-k recommendation. In this case, multiple positive samples and many negative samples are used at the same time. Therefore, we term it multisample-based contrastive loss, which is defined as follows:

$$L_{MSCL} = \sum_{m=1}^{M} L_{ICL}^{m} \tag{11}$$

Their combination forms a logic for this paper: using multiple (positive and negative) samples and solving the problems that exist in them. The proposed function is simple and effective. Two hyperparameters are introduced, but they are easy to tune.

C. Model Prediction

The model prediction is defined as the inner product of the user and item final embeddings:

$$\hat{y}_{ui} = \boldsymbol{e}_u^T \boldsymbol{e}_i \tag{12}$$

Based on this prediction, the top-k most similar items are recommended to the user.

The proposed MSCL is used for model training, and the method is named sLightGCN_MSCL. MSCL replaces the BPR loss that is used in the original LightGCN. Except for the proposed loss function, our method remains the same as LightGCN. The L2 regularization for all parameters is also used following LightGCN, and it is omitted here for clarity.

D. Model Analyses

1) Complexity Analyses: In this subsection, we analyze the complexity of sLightGCN_MSCL following SGL [36]. Since sLightGCN_MSCL does not introduce trainable parameters and there is no change in model prediction, the spatial complexity and the time complexity of the model inference are the same as those in LightGCN. The complexity of sLightGCN_MSCL can be divided into two parts, that of sLightGCN and MSCL, and they are $O(2|E|) + O(2|E|Lds|E|/N)$ and $O(mN|E|ds)$, respectively, where E is the edge in the user-item interaction graph, L, s, m denotes the number of GCN layers, the number of epochs, the number of multiple positive samples, respectively, and d, N denotes the embedding size and the batch size, respectively. For comparison, that of the BPR loss is $O(2|E|ds)$.

In fact, the overall amount of calculation is significantly reduced because the number of training epochs is substantially reduced due to better convergence performance, as shown in the training efficiency in Section V. The MSCL is $O(mN/2)$ times larger than the computational cost of BPR, but this is a simple inner product that is directly accelerated by matrix operations through the GPU. Therefore, there is no significant increase in training time in each epoch, as seen in Section V.

TABLE I
STATISTICS OF THE DATASETS

Dataset	Users	Items	Interactions	Density
Yelp2018	31,668	38,048	1,561,406	0.00130
Amazon-Book	52,643	91,599	2,984,108	0.00062
Alibaba-iFashion	300,000	81,614	1,607,813	0.00007

2) Pros and Cons: Pros: The proposed loss function is simple and easy to implement. It should be noted that our approach is model-agnostic and can be applied to many recommendation system methods. Compared to the single sampling of BPR, CL is multisampling, which is more in line with the reality that people tend to face multiple options instead of the either-or situation [48]. Theoretical analysis shows that CL has the ability to mine hard samples, which intrinsically facilitates the model optimization and training efficiency [36], [48]. Additionally, MSCL distinguishes the importance of positive and negative samples by weighting, and makes better use of limited positive samples. These improvements make MSCL better than existing multiple-sampling based contrastive learning methods. Cons: Its shortcoming is the introduction of some new parameters which need to be adjusted.

IV. EXPERIMENTS

We first introduce the basic information related to the experiments, such as datasets, evaluation metrics, and hyperparameter settings. sLightGCN_MSCL is compared with many strong baselines. We conduct ablation studies to verify the effectiveness of the proposed improvements. The main hyperparameters of sLightGCN_MSCL are discussed in detail.

A. Datasets

To evaluate the effectiveness of MSCL, we conduct experiments on three benchmark datasets: Yelp2018 [36], [56], Amazon-Book [36], [56], and Alibaba-iFashion [2], [36].

Yelp2018: Yelp2018 was adopted from the 2018 edition of the Yelp challenge. Local businesses such as restaurants and bars are viewed as items.

Amazon-Book: Amazon-review is a widely used dataset for product recommendation, and Amazon-Book from the collection is selected.

Alibaba-iFashion: Alibaba-iFashion is a large and rich dataset for fashion outfit recommendation. 300 k users and all their interactions over the fashion outfits are randomly sampled by SGL [36]. It is quite sparse, which is a significant difference from the first two datasets.

Three datasets vary significantly in domains, size, and sparsity. The statistics of the processed datasets are summarized in Table I. For comparison purposes, we directly use the split data provided in SGL [36].

B. Evaluation Metrics

Following NGCF, LightGCN, and SGL [23], [36], [56], two widely used evaluation metrics, Recall@K and NDCG@K, where K=20, are used to evaluate the performance of top-k recommendations. Recall measures the number of items that the user likes in the test data that has been successfully predicted in the top-k ranking list. NDCG considers the positions of the items, and higher scores are given if the items are ranked higher. It is a metric of ranking and thus is important for the top-k recommendation. The larger the values are, the better the performance for both metrics.

C. Hyperparameter Settings

We implement our proposed method[1] on top of the official code of LightGCN[2] based on PyTorch. We replace the loss function and follow LightGCN's settings as much as possible. The embedding size is fixed to 64, and the default batch size is 2048 for all models. The learning rate and L2 regularization coefficients are chosen by grid search in the range of $\{0.0001, 0.001, 0.01\}$ and $\{1e-5, 1e-4, \ldots, 1e-2\}$. These are hyperparameters of the original LightGCN. We adjust the hyperparameters of MSCL, M and τ, in the ranges $\{1, 3, 5, \ldots, 15\}$ and $\{0.1, 0.2, 0.5, 1.0\}$, respectively. In addition, τ is usually 0.1 or 0.2. The weight α is adjusted in [0.4,0.7].

D. Compared Methods

To demonstrate the performance of our method, we select many strong baselines for comparison. NGCF [23], LR-GCCF [24], and LightGCN [56] are recently competing baselines with GCN for top-k recommendation and have been shown to outperform several methods, including GC-MC [57], PinSage [58], and NeuMF [10]. The latest method SGL [36], which is a self-supervised-based method, is also selected. In addition, the basic method and the variable autoencoder-based methods, MF and Mult-VAE, are compared.

MF: This is a traditional method based on matrix factorization that is based only on the embeddings of users and items, namely, e_u and e_i, respectively.

NGCF [23]: NGCF integrates the bipartite graph structure into the embedding process based on the graph convolutional network. It explicitly exploits the collaborative signal in the form of high-order connectivities by propagating embeddings on the graph structure.

LR-GCCF [24]: This method enhances the recommendation performance with less complexity by removing the nonlinearity. The final embeddings are the same as NGCF.

LightGCN and sLightGCN [56]: LightGCN is the state-of-the-art GCN-based collaborative filtering model, and sLightGCN is a variant. They are described in detail in Section III.

Mult-VAE [59]: Mult-VAE extends variational autoencoders (VAEs) to collaborative filtering and uses a multinomial likelihood for the data distribution. In addition, it introduces an additional regularization parameter for optimization. It can be seen as a special case of self-supervised learning (SSL) for recommendation.

[1] [Online]. Available: https://github.com/haotangxjtu/MSCL

[2] [Online]. Available: https://github.com/gusye1234/LightGCN-PyTorch

TABLE II
OVERALL PERFORMANCE COMPARISON

Method	Yelp2018		Amazon-Book		Alibaba-iFashion	
	Recall	NDCG	Recall	NDCG	Recall	NDCG
MF	0.0441	0.0353	0.0329	0.0249	0.1020	0.0474
NGCF	0.0579	0.0477	0.0344	0.0263	0.1043	0.0486
LR-GCCF	0.0591	0.0485	0.0378	0.0292	0.1110	0.0529
LightGCN	0.0639	0.0525	0.0411	0.0315	0.1078	0.0507
sLightGCN	0.0649	0.0525	0.0469	0.0363	0.1160	0.0553
Mult-VAE	0.0584	0.0450	0.0407	0.0315	0.1041	0.0497
SGL	0.0675	0.0555	0.0478	0.0379	0.1126	0.0538
LightGCN_MSCL(ours)	0.0681	0.0564	0.0500	0.0391	0.1144	0.0546
sLightGCN_MSCL(ours)	**0.0691**	**0.0568**	**0.0580**	**0.0466**	**0.1201**	**0.0578**

SGL [36]: SGL is the latest baseline for top-k recommendations. It introduces self-supervised learning into the recommendation system based on the contrastive learning framework. It is implemented on LightGCN and uses a multitask approach that unites the contrastive loss and the BPR loss function. SGL mainly benefits from graph contrastive learning to reinforce user and item representations. Following the paper, the edge drop-based SGL that achieves the best performance is adopted here.

E. Performance Comparison

The performance comparison on the three datasets is shown in Table II. The best results are shown in bold, while underlined scores are the second best. We follow the experimental results of SGL [36], except for MF, LR-GCCF, and our methods. After statistical analysis, the standard deviations on Recall and NDCG are not larger than ± 0.0002 under different initialization seeds. We have the following observations:

MF is the most basic and simplest method and performs the worst. NGCF, LR-GCCF, and LightGCN are GCN-based methods. NGCF achieved improvements relative to MF by introducing the GCN method into top-k recommendations, especially on the Yelp2008 dataset. LR-GCCF, LightGCN and sLightGCN can be seen as improvements of NGCF. Their performances are better than NGCF, and these results are consistent with the performance in the original paper. These three methods show the significant role of graph convolution methods in recommendation systems. LightGCN is the strongest baseline and becomes the basis for subsequent methods, such as SGL and our method. LightGCN removes the nonlinear activation layer and learning parameters, making the model more applicable to recommendation systems rather than simply employing GCN, which illustrates that the GCN method should be modified to fit the recommendation system.

Mult-VAE and SGL are methods that belong to self-supervised learning (SSL). The results of Mult-VAE are generally better than those of NGCF, indicating that the variational autoencoder-based method and self-supervised learning are competitive for recommendation. The results of SGL show that it has a clear boost compared with LightGCN, and suboptimal results are obtained on two datasets, which demonstrates the advancement of contrastive learning methods.

The proposed sLightGCN_MSCL is the best among all methods. LightGCN_MSCL is also listed as a variant of our method, which is also superior to other methods. Compared to the latest and best SGL methods, the improvements of sLightGCN_MSCL on Yelp2018, Amazon-Book, and Alibaba-iFashion are 2.37%, 21.34%, 6.67% on Recall, 2.34%, 22.96%, and 7.43% on NDCG, respectively. SGL uses CL and BPR jointly in the multitask learning approach without exploiting the potential of CL. Our approach is simpler and consumes less time, which can be seen in the training efficiency in Section V. This shows the correctness of improving CL.

F. Ablation Study

MSCL combines two components: the different importance values of positive and negative samples and the use of multiple positive samples. ICL and MCL denote importance-aware CL and multiple positive sample-based CL, respectively. They are shown in Equations (8)-(10).

1) The Effectiveness of the two Components: Detailed ablation studies demonstrate the effectiveness of our two components, as shown in Table III. The comparison of CL and ICL, MCL and MSCL shows the effectiveness of adding weights to distinguish the importance of positive and negative samples. The comparison of CL and MCL and ICL and MSCL illustrates the effectiveness of data augmentation based on multiple positive samples. All the results, the two evaluation metrics on three datasets in these four comparison groups, in Table III consistently demonstrate the effectiveness of the two components.

In addition, we find that the three datasets perform differently on the two components. Amazon-Book benefits more from adding weights to distinguish the importance, while the other two datasets improve more significantly on multiple positive samples. This shows that the proposed two components are effective but perform differently depending on the dataset.

2) Detailed Analysis of the Role of Multiple Positive Samples: More comparisons in training tend to yield better results in contrastive learning, such as a large batch size. Our data augmentation approach of using multiple positive samples also increases the number of comparisons in each epoch. Therefore, one of the reasons for the good performance of multiple positive samples also involves more comparisons. However, we want to show that our proposed approach makes better use of positive

TABLE III
ABLATION STUDY

Loss	Importance-aware	Multipositive samples	Yelp2018		Amazon-Book		Alibaba-iFashion	
			Recall	NDCG	Recall	NDCG	Recall	NDCG
CL			0.0655	0.0541	0.0480	0.0399	0.1152	0.0556
ICL	✓		0.0668	0.0548	0.0544	0.0437	0.1165	0.0558
MCL		✓	0.0677	0.0559	0.0516	0.0425	0.1184	0.0573
MSCL	✓	✓	0.0691	0.0568	0.0580	0.0466	0.1201	0.0578

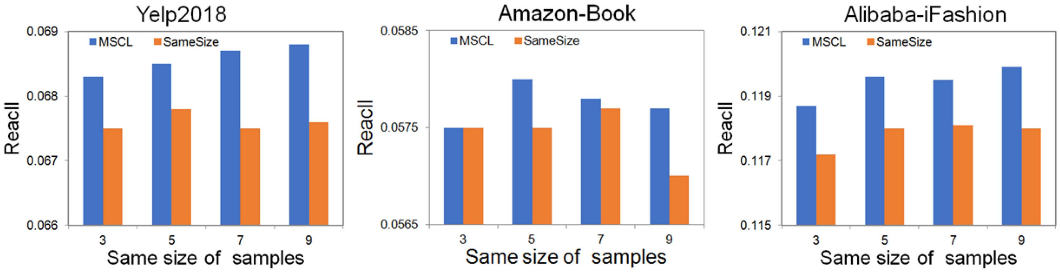


Fig. 3. Detailed analysis about the role of multiple positive samples. The horizontal axis, the same size of samples, refers to M*N samples, where N is the batch size. So the two methods in the figure have the same size of samples for comparisons. MSCL is still better than the Samesize method which does not use multiple positive samples. This indicates MSCL makes better use of the limited number of positive samples.

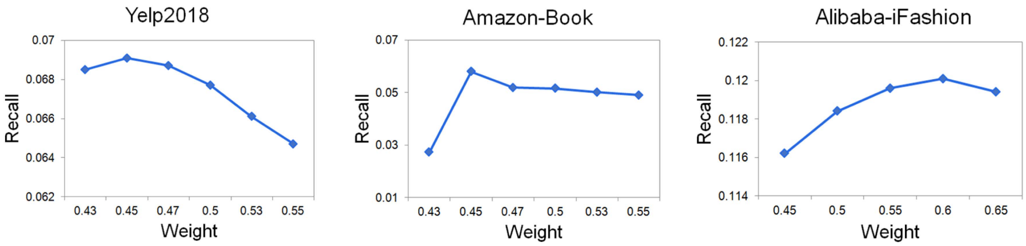


Fig. 4. Impact of the weight α. Different datasets have different optimal weights. Thousands of negative samples in the first two datasets should be given more weight. Alibaba-iFashion is too sparse, and thus, the positive samples are more important.

samples, except for the number of comparisons. Experiments with the same number of comparisons need to be performed to exclude this factor. We expand the batch size of ICL to $M*N$ because a user is compared with $M*N$ items in MSCL, where N is the training batch size.

The results are shown in Fig. 3. Our method consistently outperforms the latter on the three datasets. Overall, the performance of the same batch size peaks and falls back as the batch size increases, especially on the Amazon-Book. They are significantly worse than the performances of multiple positive samples when M=9. Yelp2018 and Alibaba-iFashion have the same trend of change in Fig. 3, which is different from that of Amazon-Book. This is consistent with the above observation in Table III that Yelp2018 and Alibaba-iFashion behave differently from Amazon-Book. In summary, the combination of multiple positive samples makes better use of the limited number of positive samples.

G. Discussion of Hyperparameters

MSCL solves the problems of CL and introduces two hyperparameters, the weight α and the number of positive samples m. So this subsection focuses on the impact of these two hyperparameters. Overall, α varies around 0.5 depending on the dataset, while m = 5 as the default setting is appropriate. Experiments show that the hyperparameters are easy to adjust.

1) Impact of the Weight: We adjust the weight α, and the results are shown in Fig. 4. The trends look somewhat different which depends on the differences of the datasets. However, the trends are similar on the whole, increasing and then decreasing. The keys of the figure are the peaks of the performance curves. The results show that both weighting methods achieve better results relative to unweighted when α is 0.5. The first two datasets both obtain the best performance at 0.45, while Alibaba-iFashion reaches the best at 0.60. The main reason for this difference is that Alibaba-iFashion is the sparsest dataset and has few positive samples of users. Users have 49.3, 56.7, and 6.4 positive items on average on the three datasets. For Yelp2018 and Amazon-Book, the imbalance problem is the main issue, and thus thousands of negative samples do require relatively more weights to learn better. Compared to the other two datasets, positive items of Alibaba-iFashion are so few that positive samples should be more important and given more weight. This illustrates that the first two datasets benefit mainly from solving the imbalance problem, and the last dataset benefits mainly from increasing the importance of a limited number of positive

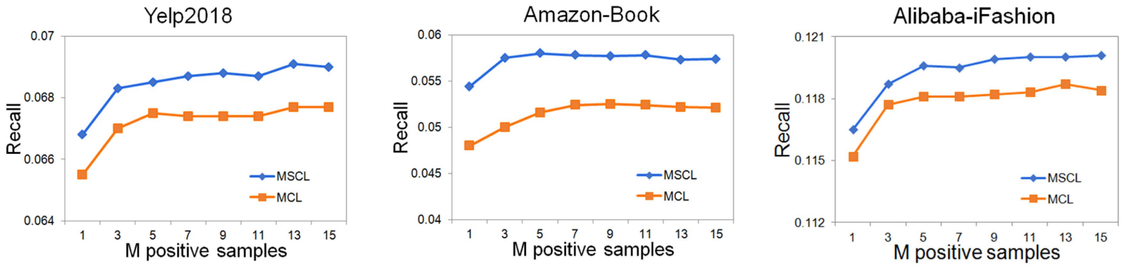


Fig. 5. Impact of the number of positive samples. The effectiveness of adding positive samples is consistently shown on both curves.

TABLE IV
PERFORMANCE OF MSCL COMPARED WITH BPR ON DIFFERENT METHODS

Method	Yelp2018		Amazon-Book		Alibaba-iFashion	
	Recall	NDCG	Recall	NDCG	Recall	NDCG
MF_BPR MF_MSCL	0.0441 0.0657(48.98%)	0.0353 0.0538(52.41%)	0.0329 0.0478(45.29%)	0.0249 0.0369(48.19%)	0.1020 0.1185(16.18%)	0.0474 0.0576(21.52%)
NGCF_BPR NGCF_MSCL	0.0579 0.0655(13.13%)	0.0477 0.0538(12.79%)	0.0344 0.0481(39.83%)	0.0263 0.0375(42.59%)	0.1043 0.1152(10.45%)	0.0486 0.0565(16.26%)
LR-GCCF_BPR LR-GCCF_MSCL	0.0591 0.0658(11.34%)	0.0485 0.0543(11.96%)	0.0378 0.0465(23.02%)	0.0292 0.0360(23.29%)	0.1072 0.1119(4.38%)	0.0507 0.0533(5.13%)
sLightGCN_BPR sLightGCN_MSCL	0.0649 **0.0691**(6.47%)	0.0525 **0.0568**(8.19%)	0.0469 **0.0580**(23.67%)	0.0363 **0.0466**(28.37%)	0.1160 **0.1201**(3.53%)	0.0553 **0.0578**(4.52%)

samples. It also demonstrates that the weighting approach can solve these two problems to balance the importance of positive and negative samples and can adapt to different datasets, despite its simplicity.

2) Impact of the Number of Positive Samples: Both MSCL and MCL are able to illustrate the role of multiple positive samples, and the results are shown in Fig. 5. All the results of MSCL and MCL with multiple positive samples are significantly better than those with only one positive sample on the left. Therefore, the proposed data augmentation method does make better use of the positive samples. MSCL and MCL have the same tendencies on the three datasets. As the number of positive samples increases, MSCL and MCL start with a significant improvement and then change flatly. It can be seen from Fig. 5 that approximately 5 or 7 is appropriate, and more positive samples tend to be slightly better. It also shows that all results of MSCL are better than those of MCL with the same number of positive samples, demonstrating the effectiveness of the proposed importance-aware loss.

V. ADVANTAGES OF MSCL

We have obtained optimal results of MSCL by the method sLightGCN_MSCL on top-k recommendation. We focus on the proposed loss function MSCL in this section. MSCL is simple and easy to implement, but it also has many other advantages, such as applicability, suitability for top-k recommendation, and high training efficiency. In addition, MSCL significantly improves the simplest and most basic model MF, making it more valuable for applications. Finally, as an extension, we verify that the problems and improvements of this paper are also generalizable to multiple sample-based BPR functions.

A. Applicability of MSCL

To show the applicability of MSCL, we apply it to many methods and compare it with the BPR loss, and methods with these two losses are named “*-MSCL” and “*-BPR”.

The results are shown in Table IV, and the percentage of improvements relative to BPR is also presented. MSCL-based methods outperform the BPR-based methods on all results on the three datasets and have significant improvements on MF, NGCF and LR-GCCF. sLightGCN_MSCL consistently obtains the best results on all datasets and has desirable improvements. In particular, the improvement on the Amazon-Book dataset is still approximately 25%.

Furthermore, MF is the most fundamental method based on embeddings in the recommendation field, and many methods can be seen as developments of MF. Theoretically, MSCL is suitable for all embedding-based methods. Thus, the effectiveness of MF shows that MSCL can be widely used in recommendation systems. The above experimental results and analysis show that our proposed MSCL is model-agnostic and widely adaptable.

B. Suitability for Top-K Recommendations

We believe that MSCL is more suitable for the top-k recommendation task. This can be illustrated by theoretical analysis and experimental results.

Theoretically, MSCL is compared with the BPR loss function. The common goal of both BPR and MSCL is to learn better feature representation by comparing positive and negative samples. BPR uses a limited number of comparisons, usually one or several, while MSCL employs thousands. Moreover, MSCL improves the quality of comparison by distinguishing the importance of positive and negative samples and makes better use of the few positive samples. MSCL makes the similarities

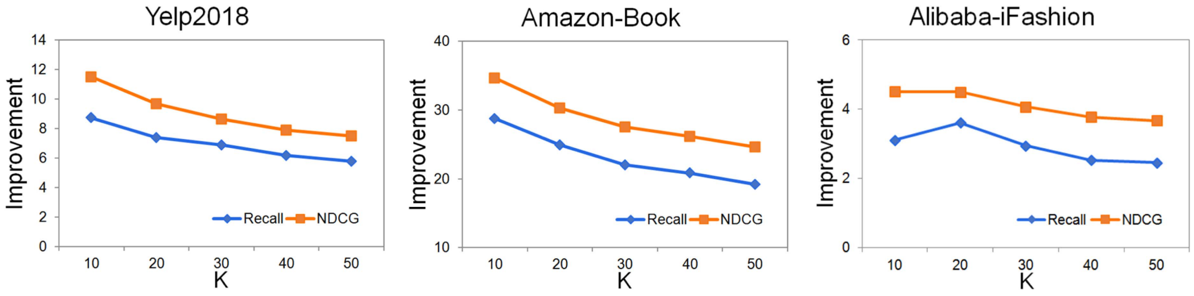


Fig. 6. Better improvements on NDCG for top-k recommendation. The figure shows the percentage improvement of MSCL over BPR on Recall@K and NDCG@K at different K. Higher improvement of NDCG@K than Recall@K shows MSCL is more suitable for top-k recommendation.

between positive and negative samples more accurate through increasingly better comparisons. The top-k recommendation is a ranking task, and BPR is proposed specifically for ranking tasks. MSCL outperforms BPR in terms of theoretical and experimental results. Therefore, MSCL can obtain better ranking results and makes more sense for top-k recommendations.

Experimentally, the improvement of the NDCG evaluation metric is more obvious. NDCG is a ranking-related metric that is more meaningful for ranking and top-k recommendation tasks than Recall. The following two observations support our conjecture well: (1) In Table IV, we found that the boost in performance by NDCG is generally more than that achieved by Recall. On average, the improvements are 19.98%, 32.95%, and 8.64% on Recall and 21.34%, 35.61%, and 11.86% on NDCG, respectively, for the three datasets. This shows the superiority of MSCL for top-k recommendation by different methods. (2) Fig. 6 shows the more significant improvement of MSCL over BPR on NDCG compared to Recall with different K. This shows that the ranking performance of NDCG is consistently higher than that of Recall even as K changes.

C. The Improvement of MSCL on MF

The performance of MF_MSCL is particularly noteworthy in Table IV.

1) MF_MSCL gains the most significant improvement among all MSCL-based versus BPR-based methods. The results are even better than those of all BPR-based methods, including sLightGCN_BPR. This indicates that MSCL with the most basic and simple method is significantly better than the excellent methods recently proposed, even the state-of-the-art GCN methods. Thus, to some extent, a good loss function works better than new models.

2) In addition, we find that the results of MF_MSCL are also competitive. They are close to or better than those of NGCF_MSCL and LR-GCCF_MSCL on all datasets and are close to sLightGCN_MSCL on the Alibaba-iFashion dataset. This indicates that MSCL is also effective in directly optimizing embeddings without a complex model, such as GCN.

These observations also indicate that MSCL can achieve competitive results in the simplest baseline, which is also consistent with the latest research Graph-MLP [60]. Graph-MLP indicates that it is sufficient to learn discriminative node representations only by implementing an MLP and graph-based CL, without the complex GCN. Compared with Graph-MLP, MF_MSCL is more concise and simple. It is based only on embeddings and improved CL functions, which is still effective even without an MLP. Furthermore, Graph-MLP does not optimize the CL loss function, which shows the great potential of MSCL.

Three other points need to be highlighted. (1) As the most basic and simple method, MF_MSCL can be widely used in various tasks of recommendation systems, not only the top-k tasks. The applicability of MSCL is best illustrated by MF_MSCL. (2) MF_MSCL also has other advantages of MSCL presented in this section, such as being more suitable for top-k recommendation and fast convergence. (3) Importantly, it is valuable for applications with high space and time requirements or industrial applications at a large scale.

D. Training Efficiency

The training efficiency of the MSCL is also significantly improved, as shown in Fig. 7. Because of the large difference in loss values, following LightGCN and SGL, the test performance on three datasets is used to show the convergence speed. In terms of the number of training epochs required to achieve optimal performance, more than 900 epochs are required for BPR, while MSCL achieves the best performance at 46, 3, and 90 epochs on the three datasets. BPR requires too many epochs for convergence, while MSCL converges earlier, so we adopt the same number of epochs as MACL for comparison.

For the first two datasets, MSCL converges directly to high values that approximate the final performance with slight fluctuations, while BPR converges slowly at lower values. On the third dataset, it is slightly more difficult to converge due to the sparsity of the dataset. The MSCL converges sharply by approximately 5 epochs to the value that approximates the final performance. This shows that MSCL has a fast convergence capability. The training efficiency is improved at least tens of times on different datasets in terms of training epochs, as mentioned before. The main reason for the high training efficiency is that multiple samples are learned at the same time, as demonstrated in [49].

Moreover, in terms of actual training time, MSCL does not significantly increase the training time per epoch. Table V shows the average time consumption in each epoch in seconds. MSCL takes less time than BPR when one positive sample is used, as shown in the first two columns of the table. BPR requires negative sampling, MSCL does not require negative sampling, and

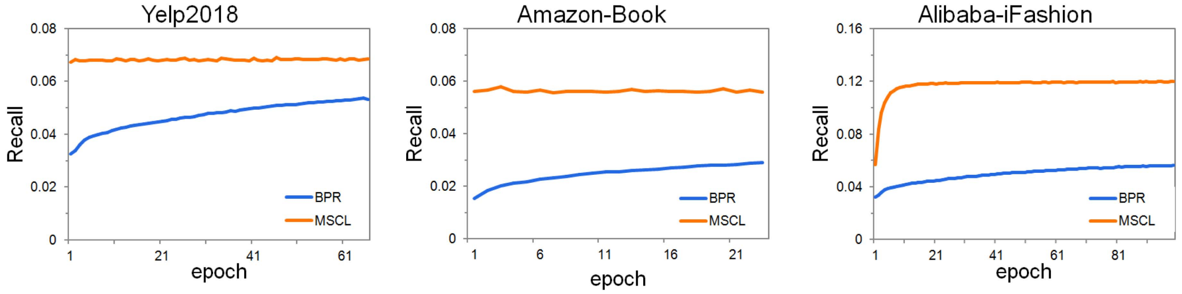


Fig. 7. Training Efficiency. Testing Recall of MSCL and BPR with sLightGCN on three datasets. Here, the total training epochs of MSCL are shown, and the curve of BPR is too long and thus shows the same training epochs as MSCL.

TABLE V
ACTUAL TRAINING TIME PER EPOCH

	BPR	MSCL	MSCL	MSCL	MSCL
		M=1	M=5	M=10	M=15
Yelp2018	13	12	15	19	22
Amazon-Book	64	61	65	71	77
Alibaba-iFashion	17	16	19	22	25

TABLE VI
PERFORMANCE COMPARISON AMONG BPR, MSBPR AND MSCL

		BPR	MSBPR	MSCL
Yelp2018	Recall	0.0649	0.0670	0.0691
	NDCG	0.0525	0.0552	0.0568
Amazon-Book	Recall	0.0469	0.0458	0.058
	NDCG	0.0363	0.0371	0.0466
Alibaba-iFashion	Recall	0.1160	0.1172	0.1201
	NDCG	0.0553	0.0564	0.0578

the computation with multiple negative samples is accelerated by the GPU. When the number of positive samples increases by 1, the average time increase on the three datasets is 0.8 s. Such time consumption is completely negligible. When M=5, MSCL and BPR consume the same time, but the performance is much better than BPR. The latest SGL [36] based on the contrastive learning framework takes approximately 3.7x longer than LightGCN, while our approach is approximately 1.5 times that of LightGCN.

The above analyses demonstrate that MSCL has remarkable improvement in convergence speed and training efficiency compared to BPR. There is no significant increase in time consumption per epoch, which is an advantage over SGL in terms of performance and time consumption.

E. Multisample-Based BPR Loss (MSBPR)

The proposed MSCL combines ICL and MCL to solve the problem of different importance values of positive and negative samples and insufficient use of positive samples. The problems and solutions are also appropriate for BPR. Therefore, we modify the loss function of the multisample-based BPR in the same way and present the MSBPR function. In this case, the same sampling method of MSCL is used by MSBPR. The formula of MSBPR is as follows:

$$L_{MSBPR} = \sum_{m=1}^{M} \sum_{(u,i)\in D} -\log\sigma\left(\alpha f\left(u,i^{+}\right)/\tau\right.$$
$$\left.-(1-\alpha) f\left(u,i^{-}\right)/\tau\right) \tag{13}$$

where σ is the logistic sigmoid. We use $f(u,i)$ instead of $\hat{y}_{ui}$ by drawing on the contrastive learning because the $\hat{y}_{ui}$-based approach does not work.

The results are shown in Table VI, and the baseline is sLightGCN. The overall MSBPR-based methods are better than BPR, while the only exception in all results is that the Recall of MSBPR is worse than BPR on Amazon-Book. It shows that our proposed idea can be extended to BPR and other pairwise-based loss functions. In addition, we find that MSCL works better than MSBPR, especially on the Amazon-Book dataset, which shows the superiority of the CL function again and the correctness of improving CL in this paper. Therefore, MSCL is better than BPR and MSBPR in the field of recommendation systems.

VI. CONCLUSION

In this paper, we propose the MSCL function for multisample-based recommendation systems. We distinguish the different importance values of positive and negative samples and propose a new data augmentation method to make better use of positive samples. MSCL is a simple approach but obtains optimal results. More importantly, it has the advantages of wide applicability to various models, suitability for top-k recommendation, and high training efficiency. MSCL makes the simple and basic MF implementation more valuable for industrial applications. These advantages make MSCL more competitive for top-k recommendation tasks.

This work represents an initial attempt to exploit improved CL for recommendations. We believe that other improvements based on CL are an important direction. The two problems, the different importance values of positive and negative samples and insufficient use of positive samples, are still valuable and deserve to be studied in depth. The proposed MSCL has the potential to be extended to the multimedia recommendation, the contrastive learning fields, as well as other fields.

REFERENCES

[1] S. Yu *et al.*, "Leveraging tripartite interaction information from live stream e-commerce for improving product recommendation," in *Proc. ACM SIGKDD Conf. Knowl. Discov. Data Mining*, Association for Computing Machinery, 2021, pp. 3886–3894.

[2] W. Chen *et al.*, "POG: Personalized outfit generation for fashion recommendation at Alibaba iFashion," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, Association for Computing Machinery, 2019, pp. 2662–2670.

[3] Z. Xu, L. Chen, Y. Dai, and G. Chen, "A dynamic topic model and matrix factorization-based travel recommendation method exploiting ubiquitous data," *IEEE Trans. Multimedia*, vol. 19, pp 1933–1945, 2017.

[4] Y. Wu, K. Li, G. Zhao, and X. QIAN, "Personalized long- and short-term preference learning for next POI recommendation," *IEEE Trans. Knowl. Data Eng.*, to be published, doi: 10.1109/TKDE.2020.3002531.

[5] G. Zhao, P. Lou, X. Qian, and X. Hou, "Personalized location recommendation by fusing sentimental and spatial context," *Knowl. Based Syst.*, vol. 196, 2020, Art. no. 105849.

[6] S. Huang, J. Zhang, L. Wang, and X.-S. Hua, "Social friend recommendation based on multiple network correlation," *IEEE Trans. Multimedia*, vol. 18, pp 287–299, 2016.

[7] G. Zhao, X. Lei, X. Qian, and T. Mei, "Exploring users' internal influence from reviews for social recommendation," *IEEE Trans. Multimedia*, vol. 21, pp 771–781, 2019.

[8] X. Chen, D. Liu, Z. Xiong, and Z.-J. Zha, "Learning and fusing multiple user interest representations for micro-video and movie recommendations," *IEEE Trans. Multimedia*, vol. 23, pp 484–496, 2021.

[9] G. Zhao, Z. Liu, Y. Chao, and X. Qian, "CAPER: Context-aware personalized emoji recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 9, pp 3160–3172, Sep. 2021.

[10] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proc. 26th Int. Conf. World Wide Web*, Association for Computing Machinery, 2017, pp. 173–182.

[11] X. He, X. Du, X. Wang, F. Tian, J. Tang, and T.-S. Chua, "Outer product-based neural collaborative filtering," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, 2018, pp. 2227–2233.

[12] Y. Liu, D. Zhang, Q. Zhang, and J. Han, "Part-object relational visual saliency," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published, doi: 10.1109/TPAMI.2021.3053577.

[13] T. Donkers, B. Loepp, and J. Ziegler, "Sequential user-based recurrent neural network recommendations," in *Proc. 11th ACM Conf. Recommender Syst.*, Association for Computing Machinery, 2017, pp. 152–160.

[14] J. Xiao, H. Ye, X. He, H. Zhang, F. Wu, and T.-S. Chua, "Attentional factorization machines: Learning the weight of feature interactions via attention networks," in *Proc. Int. Joint Conf. Artif. Intell.*, C. Sierra, Ed., 2017, pp. 3119–3125.

[15] J. Hao, Y. Dun, G. Zhao, Y. Wu, and X. Qian, "Annular-graph attention model for personalized sequential recommendation," *IEEE Trans. Multimedia*, to be published, doi: 10.1109/TMM.2021.3097186.

[16] H. Tang, G. Zhao, X. Bu, and X. Qian, "Dynamic evolution of multi-graph based collaborative filtering for recommendation systems," *Knowl.-Based Syst.*, vol. 228, 2021, Art. no. 107251.

[17] Y. Wei, X. Wang, X. He, L. Nie, Y. Rui, and T.-S. Chua, "Hierarchical user intent graph network for multimedia recommendation," *IEEE Trans. Multimedia*, to be published, doi: 10.1109/TMM.2021.3088307.

[18] Y. Wei, X. Wang, L. Nie, X. He, R. Hong, and T.-S. Chua, "MMGCN: Multi-modal graph convolution network for personalized recommendation of micro-video," in *Proc. ACM Multimedia*, Association for Computing Machinery, 2019, pp. 1437–1445.

[19] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," in *Proc. 25th Conf. Uncertainty Artif. Intell.*, 2009, pp. 452–461.

[20] X. Lei, X. Qian, and G. Zhao, "Rating prediction based on social sentiment from textual reviews," *IEEE Trans. Multimedia*, vol. 18, pp. 1910–1921, 2016.

[21] G. Zhao, X. Qian, and X. Xie, "User-service rating prediction by exploring social users' rating behaviors," *IEEE Trans. Multimedia*, vol. 18, pp. 496–506, 2016.

[22] G. Zhao, X. Qian, X. Lei, and T. Mei, "Service quality evaluation by exploring social users' contextual information," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 12, pp. 3382–3394, Dec. 2016.

[23] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2019, pp. 165–174.

[24] L. Chen, L. Wu, R. Hong, K. Zhang, and M. Wang, "Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach," in *Proc. AAAI Conf. Artif. Intell.*, AAAI Press, 2020, pp. 27–34.

[25] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, "A Simple Framework for Contrastive Learning of Visual Representations. in *Proc. Int. Conf. Mach. Learn.*, Proceedings of Machine Learning Research, 2020, vol. 119, pp. 1597–1607.

[26] K. He, H. Fan, Y. Wu, S. Xie, and Ross B. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9726–9735.

[27] P. Khosla *et al.*, " Supervised contrastive learning," in *Proc. Conf. Neural Inf. Process. Syst.*, 2020, pp 18661–18673.

[28] J. Li, P. Zhou, C. Xiong, and S. C. H. Hoi, "Prototypical contrastive learning of unsupervised representations," in *Proc. Int. Conf. Learn. Representations*, 2021, https://openreview.net/forum?id=KmykpuSrjcq.

[29] J.-B. Grill *et al.*, "Bootstrap your own latent - A new approach to self-supervised learning," in *Proc. Conf. Neural Inf. Process. Syst.*, 2020.

[30] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, "Graph contrastive learning with augmentations," in *Proc. Conf. Neural Inf. Process. Syst.*, 2020, pp. 5812–5823.

[31] G. Wu *et al.*, "Unsupervised deep video hashing via balanced code for large-scale video retrieval," *IEEE Trans. Image Process.*, vol. 28 no. 4, pp. 1993–2007, Apr. 2019.

[32] Y. Miao, Z. Lin, X. Ma, G. Ding, and J. Han, "Learning transformation-invariant local descriptors with low-coupling binary codes," *IEEE Trans. Image Process.*, vol. 30, pp. 7554–7566, 2021.

[33] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2005, vol. 1, pp. 539–546.

[34] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2006, pp. 1735–1742.

[35] X. Song and Z. Jin, "Robust label rectifying with consistent contrastive-learning for domain adaptive person re-identification," *IEEE Trans. Multimedia*, to be published, doi: 10.1109/TMM.2021.3096014.

[36] J. Wu *et al.*, "Self-supervised graph learning for recommendation," in *Proc. 44th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, Association for Computing Machinery, 2021, pp. 726–735.

[37] X. Xie, F. Sun, Z. Liu, J. Gao, B. Ding, and B. Cui, "Contrastive pre-training for sequential recommendation," *CoRR*, 2020, *arXiv:2010.14395*.

[38] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 815–823.

[39] G. Chechik, V. Sharma, U. Shalit, and S. Bengio, "Large scale online learning of image similarity through ranking," *J. Mach. Learn. Res.*, vol. 11, pp. 1109–1135, 2010.

[40] K. Q. Weinberger, J. Blitzer, and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," in *Proc. Neural Inf. Process. Syst.*, 2005, pp. 1473–1480.

[41] K. Sohn, "Improved deep metric learning with multi-class N-pair loss objective," in *Proc. Neural Inf. Process. Syst.*, 2016, pp. 1857–1865.

[42] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *CoRR*, 2018, *arXiv:1807.03748*.

[43] R. D. Hjelm *et al.*, "Learning deep representations by mutual information estimation and maximization," in *Proc. Int. Conf. Learn. Representations*, 2019, https://openreview.net/forum?id=Bklr3j0cKX.

[44] P. Bachman, R. D. Hjelm, and W. Buchwalter, "Learning representations by maximizing mutual information across views," in *Proc. Neural Inf. Process. Syst.*, 2019, pp. 15509–15519.

[45] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3733–3742.

[46] X. Tong, P. Wang, C. Li, L. Xia, and S.-Z. Niu, "Pattern-enhanced contrastive policy learning network for sequential recommendation," in *Proc. Int. Joint Conf. Artif. Intell.*, 2021, pp. 1593–1599.

[47] Z. Xie, C. Liu, Y. Zhang, H. Lu, D. Wang, and Y. Ding, "Adversarial and contrastive variational autoencoder for sequential recommendation," in *Proc. World Wide Web Conf.*, ACM / IW3C2, 2021, pp. 449–459.

[48] Y. Wei *et al.*, "Contrastive learning for cold-start recommendation," in *Proc. ACM Int. Conf. Multimedia*, 2021, pp. 5382–5390.

[49] T. Chen, Y. Sun, Y. Shi, and L. Hong, "On sampling strategies for neural network-based collaborative filtering," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2017, pp. 767–776.

[50] J. Qiu *et al.*, “GCC: Graph contrastive coding for graph neural network pre-training,” in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, Association for Computing Machinery, 2020, pp. 1150–1160.

[51] K. Hassani and A. H. K. Ahmadi, “Contrastive Multi-View Representation Learning on Graphs,” in *Proc. Int. Conf. Mach. Learn.*, Proceedings of Machine Learning Research, 2020, vol. 119, pp. 4116–4126.

[52] Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, and L. Wang, “Graph contrastive learning with adaptive augmentation,” in *Proc. World Wide Web Conf.*, ACM / IW3C2, 2021, pp. 2069–2080.

[53] T. Zhao, Y. Liu, L. Neves, Oliver J. Woodford, M. Jiang, and N. Shah, “Data augmentation for graph neural networks,” in *Proc. AAAI Conf. Artif. Intell.*, AAAI Press, 2021, pp. 11015–11023.

[54] J. Thoma, D.P. Paudel, and L.V. Gool, “Soft contrastive learning for visual localization,” in *Proc. Neural Inf. Process. Syst.*, 2020, pp. 11119–11130.

[55] C.-Y. Chuang, J. Robinson, Y.-C. Lin, A. Torralba, and S. Jegelka, “Debiased contrastive learning,” in *Proc. Neural Inf. Process. Syst.*, 2020, pp 8765–8775.

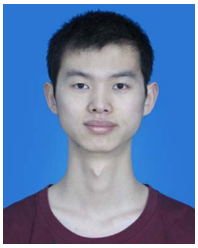
[56] X. He, K. Deng, X. Wang, Y. Li, Y.-D. Zhang, and M. Wang, “LightGCN: Simplifying and powering graph convolution network for recommendation,” in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, Association for Computing Machinery, 2020, pp. 639–648.

[57] R. van den Berg, T. N. Kipf, and M. Welling, “Graph convolutional matrix completion,” *CoRR*, 2017, *arXiv:1706.02263*.

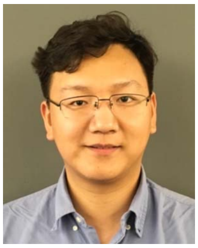
[58] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, “Graph convolutional neural networks for web-scale recommender systems,” in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, pp. 974–983, 2018.

[59] D. Liang, R. G. Krishnan,M. D. Hoffman, and T. Jebara, “Variational autoencoders for collaborative filtering,” in *Proc. World Wide Web Conf.*, Association for Computing Machinery, 2018, pp. 689–698.

[60] Y. Hu, H. You, Z. Wang, Z. Wang, E. Zhou, and Y. Gao, Graph-MLP: Node Classification without Message Passing in Graph,” *CoRR*, 2021, *arXiv:2106.04051*.



Hao Tang received the B.E. degree from The PLA Information Engineering University, Zhengzhou, China, in 2011 and the M.E. degree from the Shandong University of Science and Technology, Qingdao, China, in 2013. After working for 5 years, he is currently working toward the Ph.D. degree with SMILES LAB, Xi’an Jiaotong University, Xi’an, China. His current research interests include recommendation systems and graph neural networks.



Guoshuai Zhao (Member, IEEE) received the B.E. degree from Heilongjiang University, Harbin, China, in 2012, and the M.S. and Ph.D. degrees from Xi’an Jiaotong University, Xi’an, China, in 2015 and 2019, respectively. He was an Intern with Social Computing Group, Microsoft Research Asia from January 2017 to July 2017 and was a Visiting Scholar with Northeastern University, Boston, MA, USA, from October 2017 to October 2018 and with the Massachusetts Institute of Technology, Cambridge, MA, USA, from June 2019 to December 2019. He is currently an Associate Professor with Xi’an Jiaotong University. His research interests include social media Big Data analysis, recommendation systems, and natural language generation.

Yuxia Wu received the B.S. degree from Zhengzhou University, Henan, China, in 2014 and the M.S. degree from Fourth Military Medical University, Xi’an, China, in 2017. She is currently working toward the Ph.D. degree with Xi’an Jiaotong University, Xi’an, China. She is mainly engaged in the research of social multimedia mining and recommendation systems.

Xueming Qian (Member, IEEE) received the B.S. and M.S. degrees from the Xi’an University of Technology, Xi’an, China, in 1999 and 2004, respectively, and the Ph.D. degree in electronics and information engineering from Xi’an Jiaotong University, Xi’an, China, in 2008. From 2010 to 2011, he was a Visiting Scholar with Microsoft Research Asia, Beijing, China. He was previously an Assistant Professor with Xi’an Jiaotong University, where he was an Associate Professor from 2011 to 2014, and is currently a Full Professor. He is also the Director ofSmiles Laboratory, Xi’an Jiaotong University. His research interests include social media Big Data mining and search.